

复杂流程设计代码说明

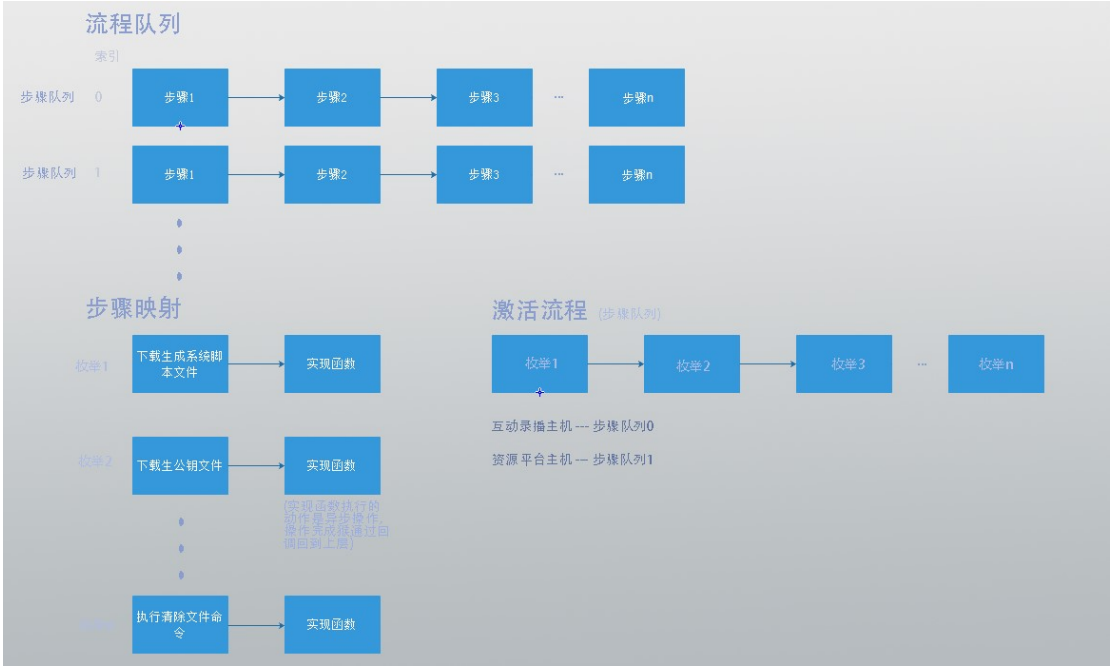
一、需求背景

录播激活设备流程实际上就是在录播主机产生一些配置文件，设置主机 WEB 服务器的一些参数以及生成授权文件；

而最初的录播激活设备流程只是生成授权文件，并将授权文件上传到录播主机上，随着录播服务的升级，此流程越来越复杂，一是步骤增减，二是交互增多；

为了方便以后的扩展而做此次设计，这样设计可以很方便调整步骤顺序和增加新的步骤和删减冗余步骤。

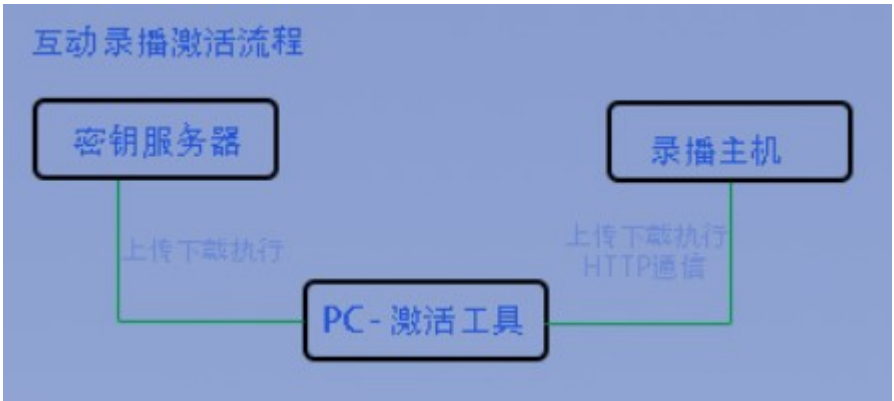
二、设计思路



三、实现说明

在具体实现层面描述实现方案（最好有类图）

- CKeyMainWnd --- 主界面类
- CBaseHttp --- HTTP 通信类
- CSshUExec --- SSH 操作类 (上传下载和执行命令)



```

298 // 初始化激活步骤映射
299 if (m_mapActiveStepFun.empty())
300 {
301     m_mapActiveStepFun[STEP_DOWN_KHOST_CLIREG] = &CKeyMainWnd::Active_Step9;
302     m_mapActiveStepFun[STEP_DOWN_KHOST_PUBKEY] = &CKeyMainWnd::Active_Step10;
303     m_mapActiveStepFun[STEP_DOWN_KHOST_KEY] = &CKeyMainWnd::Active_Step4;//
304     m_mapActiveStepFun[STEP_EXEC_KHOST_LICENSE] = &CKeyMainWnd::Active_Step3;//
305     m_mapActiveStepFun[STEP_UP_KHOST_REG] = &CKeyMainWnd::Active_Step2;//
306     m_mapActiveStepFun[STEP_DOWN_AHOST_REG] = &CKeyMainWnd::Active_Step1;//
307     m_mapActiveStepFun[STEP_EXEC_AHOST_CLIREG] = &CKeyMainWnd::Active_Step13;
308     m_mapActiveStepFun[STEP_EXEC_AHOST_CLEAR] = &CKeyMainWnd::Active_Step14;
309     m_mapActiveStepFun[STEP_UP_AHOST_CLIREG] = &CKeyMainWnd::Active_Step11;
310     m_mapActiveStepFun[STEP_UP_AHOST_PUBKEY] = &CKeyMainWnd::Active_Step12;
311     m_mapActiveStepFun[STEP_UP_AHOST_KEY] = &CKeyMainWnd::Active_Step5;//
312     m_mapActiveStepFun[STEP_UP_AHOST_VIDEOSTRATEGY] = &CKeyMainWnd::Active_Step6;//
313     m_mapActiveStepFun[STEP_UP_AHOST_SOFTDETECTION] = &CKeyMainWnd::Active_Step7;//
314     m_mapActiveStepFun[STEP_WEB_SEND_STREAMS] = &CKeyMainWnd::Active_Step0;//
315     m_mapActiveStepFun[STEP_END] = nullptr;//
316 }
317 if (m_mapActiveStepScheme.empty())
318 {
319     uint32_t uIndex = 0;
320     ActiveStepScheme activeStepSchemeTmp;
321     activeStepSchemeTmp.push_back(STEP_START);
322     activeStepSchemeTmp.push_back(STEP_DOWN_KHOST_CLIREG);
323     activeStepSchemeTmp.push_back(STEP_DOWN_KHOST_PUBKEY);
324     activeStepSchemeTmp.push_back(STEP_UP_AHOST_CLIREG);
325     activeStepSchemeTmp.push_back(STEP_UP_AHOST_PUBKEY);
326     activeStepSchemeTmp.push_back(STEP_WEB_SEND_STREAMS);
327     activeStepSchemeTmp.push_back(STEP_EXEC_AHOST_CLIREG);
328     activeStepSchemeTmp.push_back(STEP_DOWN_AHOST_REG);
329     activeStepSchemeTmp.push_back(STEP_UP_KHOST_REG);
330     activeStepSchemeTmp.push_back(STEP_EXEC_KHOST_LICENSE);
331     activeStepSchemeTmp.push_back(STEP_DOWN_KHOST_KEY);
332     activeStepSchemeTmp.push_back(STEP_UP_AHOST_KEY);
333     activeStepSchemeTmp.push_back(STEP_UP_AHOST_VIDEOSTRATEGY);
334     activeStepSchemeTmp.push_back(STEP_UP_AHOST_SOFTDETECTION);
335     activeStepSchemeTmp.push_back(STEP_EXEC_AHOST_CLEAR);
336     activeStepSchemeTmp.push_back(STEP_END);
337     m_mapActiveStepScheme[uIndex] = activeStepSchemeTmp;
338     uIndex++;
339 }

```

```

25 // DOWN - 下载 , UP - 上传 , AHOST - 需激活主机 , KHOST - 密钥主机
26 enum Active_Step
27 {
28     STEP_START = 0,
29     // 密钥主机操作
30     STEP_DOWN_KHOST_CLIREG = 11, // 下载CliReg文件
31     STEP_DOWN_KHOST_PUBKEY = 12, // 下载公钥文件
32     STEP_DOWN_KHOST_KEY = 13, // 下载Key文件
33     STEP_EXEC_KHOST_LICENSE = 14, // 执行脚本, 生成Key文件
34     STEP_UP_KHOST_REG = 15, // 上传Reg文件
35     // 激活主机操作
36     STEP_DOWN_AHOST_REG = 21, // 下载Reg文件
37     STEP_EXEC_AHOST_CLIREG = 22, // 执行脚本, 生成Reg文件
38     STEP_EXEC_AHOST_CLEAR = 23, // 清理临时文件
39     STEP_UP_AHOST_CLIREG = 24, // 上传CliReg文件
40     STEP_UP_AHOST_PUBKEY = 25, // 上传CliReg文件
41     STEP_UP_AHOST_KEY = 26, // 上传Key文件
42     STEP_UP_AHOST_VIDEOSTRATEGY = 27, // 上传videostategy文件
43     STEP_UP_AHOST_SOFTDETECTION = 28, // 上传softdetection文件
44     STEP_WEB_SEND_STREAMS = 29, // 设置流信息
45     // 结束
46     STEP_END = 100,
47     // 下一步
48     STEP_NEXT = 200,
49 };// step 11-12-24-25-29-22-21-15-14-13-26-27-28

```

```
444         SetCurActiveStep(SETP_NEXT);
445         emResult = Active_ExecCurStep();

107         void OnNextStepCallback(int nResult, const char *pData, int nDataLen, void *pExtendData);

77         void OnReportError(int nResult);
78         void OnActiveSuccess();
```

四、代码文件

KeyMainWnd.h 和 KeyMainWnd.cpp 主界面实现类文件

激活按钮 ---> Click 通知 ---> OnActiveBtn() ----> 流程开始

---> 成功 OnActiveSuccess()

---> 失败 OnReportError()

五、总结

这种设计后就不用关注流程有多复杂，只需要关注每个步骤点，方便合理规划步骤的顺序，删减步骤，增加步骤也只需要写一个步骤实现函数，加入映射，然后添加到步骤队列中即可。